

РАЗРАБОТКА И РЕАЛИЗАЦИЯ АВТОНОМНОГО УЗЛА ВИДЕОНАБЛЮДЕНИЯ НА БАЗЕ МИКРОКОМПЬЮТЕРА

В работе рассматривается процесс проектирования и создания полностью автономного локального сервера на базе микрокомпьютера, предназначенного для трансляции, обработки и архивирования видеопотока без использования сторонних облачных сервисов. Проведен глубокий анализ протоколов передачи видео (RTSP, MJPEG, HLS) и программных архитектур захвата данных. Особое внимание уделено вопросам аппаратной интеграции, включая систему активного охлаждения процессора и обеспечение кибербезопасности через изоляцию локальной сети. Результатом работы является функционирующий узел, обеспечивающий приватность данных и отсутствие финансовых затрат на облачные подписки.

Введение

В современных реалиях обеспечение безопасности объектов неразрывно связано с использованием систем видеонаблюдения. Однако подавляющее большинство коммерческих решений опирается на облачную инфраструктуру, что порождает две критические проблемы. Во-первых, возникает риск компрометации частных данных, так как видеопоток обрабатывается и хранится на сторонних серверах. Во-вторых, пользователь сталкивается с постоянными финансовыми издержками в виде оплаты облачных подписок. В связи с этим разработка полностью автономных, локальных систем видеонаблюдения на базе доступных микрокомпьютеров является высокоактуальной задачей.

Цель работы заключается в проектировании и создании автономного локального сервера на базе микрокомпьютера для бесперебойной трансляции, обработки и архивирования видеопотока без использования сторонних облачных сервисов.

1. Теоретические основы

Для корректного проектирования системы был проведен анализ всего пути прохождения видеосигнала: от матрицы объектива камеры до конечного устройства отображения.

В рамках исследования были проанализированы основные протоколы передачи потоковых данных:

- RTSP (Real Time Streaming Protocol): Является отраслевым «золотым стандартом». Обеспечивает прямое управление потоком и отличается минимальной задержкой.
- HTTP/MJPEG: Метод трансляции последовательности независимых JPEG-изображений. Не требует мощностей для декодирования, но потребляет значительную пропускную способность.
- HLS/DASH: Видеопоток фрагментируется на сегменты. Высокая кроссплатформенность, но существенная задержка (до нескольких секунд).

Обоснование выбора: В реализованной системе был применен гибридный подход. Для внутреннего процессинга выбран RTSP (H.264) для минимальной задержки. Для веб-интерфейса использовался HTTP/MJPEG, что позволило снизить нагрузку на клиентские устройства и обеспечить отображение в любом браузере без плагинов. Итоговая архитектура представлена на рисунке 1.



Рис. 1 – Архитектура конвейера обработки видеоданных

Функционирование программного комплекса построено по принципу конвейера (Pipeline):

- Захват (Capture): Получение данных через подсистему ядра Linux — V4L2 (Video4Linux2).
- Кодирование (Encoding): Реализовано сжатие в H.264 с задействованием аппаратной акселерации SoC.
- Детекция (Motion Detection): Алгоритм кадрового сравнения текущего и опорного кадров с учетом цифрового шума.

II. Программная реализация

Ядром системы стала программная среда, отвечающая за маршрутизацию потоков и логику системы безопасности.

Были рассмотрены три основные архитектуры: ZoneMinder (отклонена из-за сложности), Go2RTC (отсутствие архивации «из коробки») и MotionEye.

Реализация: Выбран MotionEye за баланс функциональности и потребления ресурсов. После развертывания ОС на базе Linux, пакет MotionEye инсталлирован как системный сервис. В веб-интерфейсе добавлен интерфейс /dev/video0, настроено разрешение 720p, заданы маски зон детекции и пороги чувствительности алгоритма Motion.

Реализована система циклической записи:

- Глубина архива: Лимит хранения файлов продолжительностью 7 дней.
- Автоматическая ротация: Алгоритм удаления старых фрагментов при достижении лимита пространства или времени.

– Индексация: Использование SQLite для хранения временных меток (timestamps) событий, что позволило создать календарь инцидентов.

III. Сборка и интеграция системы

Аппаратная реализация потребовала решения задач для режима 24/7.

– Интерфейсы: Использована внешняя USB-камера (UVC-драйверы) для упрощения замены сенсора.

– Охлаждение: Спроектирована двухконтурная система: массивный алюминиевый радиатор (пассивный) и высокооборотистый вентилятор (активный) для исключения троттлинга процессора при 100% нагрузке.

– Питание: Стабилизированный источник 5V/3A для защиты файловой системы и питания периферии.

– Ядро: Основной «компьютер» системы – Raspberry Pi 4B, который отвечает полностью за связь всех компонентов, а также логику системы.

В итоге собрана система, показанная на рисунке 2. Она и выполняет поставленную задачу.



Рис. 2 – Аппаратная реализация узла с системой охлаждения

IV. Организация удаленного доступа

Система спроектирована с приоритетом сетевой изоляции. Удаленный доступ реализован исключительно локально через статический IP-адрес (например, 192.168.1.x) внутри домашней подсети. Проброс портов в Internet не производился, что свело к нулю векторы внешних кибератак и обеспечило абсолютную приватность данных.

Заключение и выводы

В ходе выполнения работы успешно разработан автономный узел видеонаблюдения. Исследование доказало, что микрокомпьютеры класса Raspberry Pi способны выполнять роль полноценного NVR. Грамотный подбор программного стека (MotionEye), использование энергоэффективных кодеков и оптимизированных алгоритмов детекции позволили нивелировать аппаратные ограничения. Реализованная система активного охлаждения обеспечила стабильность работы. Итоговый продукт исключает зависимость от платных подписок и гарантирует 100% конфиденциальность за счет локальной маршрутизации данных.

Список литературы

1. Петин, В. А. Микрокомпьютеры Raspberry Pi. Практическое руководство / В. А. Петин. – СПб. : БХВ-Петербург, 2015. – 240 с.
2. Pajankar, A. Raspberry Pi Computer Vision Programming / A. Pajankar. – 2nd ed. – Birmingham: Packt Publishing, 2020. – 322 p.
3. Кулаков, Ю. А. Компьютерные сети. Протоколы, стандарты, интерфейсы / Ю. А. Кулаков, А. С. Омельченко. – К. : Юниор, 2005. – 400 с.

Василевский Владислав Валерьевич, студент кафедры информационных технологий автоматизированных систем, БГУИР, vlad.vasilevskiy.07@gmail.com.

Рында Роман Дмитриевич, студент кафедры информационных технологий автоматизированных систем, БГУИР, romaryndaofficial@gmail.com.

Научный руководитель: Шилин Леонид Юрьевич, доктор технических наук, профессор кафедры ИТАС, БГУИР, shilin@bsuir.by.